บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้อง

ในบทนี้ได้กล่าวถึง โปรโตคอลทีซีพี/ไอพี และหลักการทำงานของโปรแกรมในลักษณะไคลเอ็นต์เซอร์ฟเวอร์ ซึ่งเป็นพื้นฐานของการติดต่อสื่อสาร ระหว่างโปรแกรมฟิงเกอร์ไคลเอ็นต์ และโปรแกรมฟิงเกอร์เซอร์ฟเวอร์ และอธิบายถึงขั้นตอนการทำงานของชุดโปรแกรมฟิงเกอร์ด้วย

ทีซีพี/ไอพี และ ไคลเอ็นต์-เซอร์ฟเวอร์

โปรโตคอล ทีซีพี/ไอพี

โปรโตคอลทีซีพี/ไอพี (TCP/IP, Transmission Control Protocol / Internet Protocol) เป็นโปรโตคอลของการสื่อสารข้อมูล ที่ถูกพัฒนาขึ้นโดย Defense Advanced Research Projects Agency (DARPA) ได้รับการยอมรับเป็นมาตรฐานในระบบการสื่อสารข้อมูล สามารถใช้ติดต่อสื่อสารกับทุก ๆ ระบบเครือข่ายที่อยู่ในกลุ่มเดียวกัน ก่อให้เกิดเป็นระบบเครือข่ายที่ใหญ่มาก คือ อินเทอร์เน็ต (Internet) ซึ่งเอื้ออำนวยให้นักวิจัยในสถาบันต่าง ๆ ที่อยู่ห่างไกลกัน สามารถเรียกใช้ หรือค้นหาข้อมูลร่วมกันได้อย่างง่ายดาย รวดเร็ว เป็นที่นิยมใช้กันอย่างแพร่หลาย โปรโตคอลทีซีพี/ไอพี เป็นโปรโตคอลมาตรฐานของการสื่อสารข้อมูลสำหรับเครื่องคอมพิวเตอร์ที่ใช้ระบบยูนิกซ์ในปัจจุบัน (Stevens,1991)

โปรโตคอลทีซีพี (TCP, transmission control protocol)

โปรโตคอลทีซีพี เทียบได้กับระดับชั้นทรานสปอร์ต (transport layer) ของมาตรฐานการสื่อสารข้อมูลโอเอสไอ (OSI model) ทีซีพี จะเป็นตัวจัดการให้บริการส่งข้อมูลแบบรับส่งได้ทั้งไปและกลับพร้อมกันอย่างเชื่อถือได้ (reliable, full duplex) แก่โปรโตคอลในระดับที่สูงขึ้นไป เอื้ออำนวยให้แต่ละโปรเซส (process) ที่อยู่บนเครื่องเดียวกัน หรือต่างเครื่อง

กันสามารถส่งผ่านข้อมูลระหว่างกันได้ โดยที่ ทีซีพี ทำงานเป็นแบบ connection-oriented หมายถึง การที่คู่ของการสื่อสารข้อมูลต้องมีการตกลงจัดตั้งช่องทางการติดต่อสื่อสารระหว่างกันให้เรียบร้อยก่อนที่จะเริ่มการสื่อสารข้อมูล และโปรโตคอลทีซีพี ทำงานโดยการส่งข้อมูลผ่านโปรโตคอลไอพี (IP, internet protocol) ที่อยู่ในระดับต่ำกว่า (Stevens, 1991)

โปรโตคอลไอพี (IP, Internet Protocol)

โปรโตคอลไอพี เทียบได้กับระดับเน็ตเวอร์ค (network layer) ของมาตรฐานการสื่อสารข้อมูลโอเอสไอ สนับสนุนการส่งผ่านข้อมูลในแบบคอนเนคชั่นเลส (connectionless) โดยการแบ่งข้อมูลออกเป็นกลุ่มข้อมูลเล็ก ๆ เรียกว่า ดาต้าแกรม (datagram) จำนวนมาก แล้วทยอยส่งออกไป โดยแต่ละกลุ่มข้อมูลอาจถูกส่งไปถึงยังจุดหมายปลายทางโดยใช้เส้นทางที่ไม่ซ้ำกัน จึงอาจทำให้มีการสูญหายของข้อมูลระหว่างทางได้ หรือมีการซ้ำซ้อน หรือได้รับข้อมูลผิดลำดับ ปัญหาเหล่านี้จะถูกควบคุมโดยโปรโตคอลที่อยู่เหนือกว่า คือ ทีซีพี (Stevens, 1991)

เนื่องจากโปรโตคอลทีซีพี และไอพี อยู่ใกล้ชิดกันมาก และต้องทำงานร่วมกันเพื่อให้เกิดการรับส่งข้อมูลที่เชื่อถือได้ (reliable) จึงเรียกรวมกันว่า โปรโตคอลทีซีพี/ไอพี (Stevens, 1991)

ไคลเอ็นต์-เซอร์ฟเวอร์ (client-server)

รูปแบบที่ใช้ในการพัฒนาโปรแกรมเกี่ยวกับระบบเครือข่ายที่นิยมใช้กัน คือ ไคลเอ็นต์-เซอร์ฟเวอร์ โดยการทำงานคือ เซอร์ฟเวอร์ จะเป็นโปรเซสที่คอยรอรับการติดต่อจากไคลเอ็นต์ เพื่อให้บริการแก่ไคลเอ็นต์ ลักษณะการทำงานของรูปแบบไคลเอ็นต์-เซอร์ฟเวอร์ มีลักษณะดังนี้

1. เมื่อระบบเริ่มต้นการทำงานของเซอร์ฟเวอร์ เซอร์ฟเวอร์จะกำหนดค่าเริ่มต้นต่าง ๆ เพื่อเตรียมการทำงาน เสร็จแล้วรอรับการติดต่อจากไคลเอ็นต์

2. โปรเซสของไคลเอ็นต์ อาจปฏิบัติงานอยู่ในระบบเดียวกัน หรือปฏิบัติงานอยู่ในระบบอื่นที่อยู่ห่างไกล ที่สามารถติดต่อกับเซอร์ฟเวอร์ได้ด้วยระบบเครือข่าย โปรเซสของไคลเอ็นต์จะส่งคำขอใช้บริการผ่านระบบเครือข่าย เพื่อขอรับบริการที่เซอร์ฟเวอร์มีให้บริการอยู่

3. เมื่อเซอร์ฟเวอร์ได้รับคำขอใช้บริการของไคลเอ็นต์ จะทำการตรวจสอบสิทธิในการใช้บริการ และให้บริการแก่ไคลเอ็นต์ จากนั้นเซอร์ฟเวอร์จะกลับเข้าสู่ภาวะรอการติดต่อจากไคลเอ็นต์อีก

โปรเซสของเซอร์เวอร์ แบ่งตามการทำงานของการให้บริการได้ 2 ประเภทดังนี้

1. อิเทอเรทีพ เซอร์ฟเวอร์ (iterative server) คือ เซอร์ฟเวอร์ที่ทำการให้บริการแก่ไคลเอ็นต์ด้วยโปรเซสของตนเอง ซึ่งมักจะใช้เวลาสั้น ๆ ในการให้บริการ

2. คอนเคอเร็นท์ เซอร์ฟเวอร์ (concurrent server) คือ เซอร์ฟเวอร์ที่ให้บริการแบบพร้อมกัน (concurrent) ทำได้โดยการที่เซอร์ฟเวอร์จะสร้างโปรเซสใหม่อีก 1 โปรเซส เพื่อเป็นผู้ให้บริการแก่ไคลเอ็นต์แทน โดยที่เซอร์ฟเวอร์จะได้กลับไปรอรับการติดต่อจากไคล์เอ็นต์อื่นอีก การให้บริการแบบนี้ใช้กับกรณีที่การบริการให้แก่ไคล์เอ็นต์นั้น ใช้เวลาไม่แน่นอน โดยทั่วไปแล้วเซอร์ฟเวอร์ประเภทนี้ต้องทำงานบนระบบปฏิบัติการที่ยอมให้โปรเซสหลาย ๆ โปรเซสทำงานไปพร้อม ๆ กันได้

โปรเซสของเซอร์ฟเวอร์ มีขั้นตอนการทำงานอย่างคร่าว ๆ ดังนี้

1. เปิดช่องทางการสื่อสาร
2. คอยรอรับการติดต่อจากไคลเอ็นต์
3. สำหรับ อิเทอเรทีพ เซอร์ฟเวอร์ มักใช้ในกรณีที่การให้บริการสามารถทำได้ด้วยหนึ่งคำสั่งจากเซอร์ฟเวอร์ สำหรับคอนเคอเร็นท์เซอร์ฟเวอร์ จะสร้างโปรเซสใหม่ เพื่อให้บริการแก่ไคลเอ็นต์ ซึ่งโปรเซสใหม่นี้จะให้บริการแก่ไคลเอ็นต์ และไม่รับการติดต่อจากไคลเอ็นต์อื่น ๆ อีก และเมื่อให้บริการเสร็จ จะเลิกการทำงานโดยปิดช่องการสื่อสารกับไคลเอ็นต์นั้น
4. กลับไปทำข้อ 2

ระหว่างขั้นตอนดังกล่าว ระบบจะจัดคิวอย่างใดอย่างหนึ่งของการขอจากไคลเอ็นต์ที่มาถึง ในขณะที่เซอร์ฟเวอร์กำลังให้บริการแก่ไคลเอ็นต์อื่นอยู่

โปรเซสของไคลเอ็นต์ มีการทำงานโดยทั่วไปดังนี้

1. เปิดช่องการสื่อสาร และขอติดต่อไปยังเซอร์ฟเวอร์
2. ส่งข้อความการขอบริการไปยังเซอร์ฟเวอร์ และรับการให้บริการ ทำเช่นนี้เรื่อยไปเท่าที่ต้องการ
3. ปิดช่องการสื่อสารกับเซอร์ฟเวอร์

โปรแกรมฟิงเกอร์

ชุดโปรแกรมฟิงเกอร์ (finger user information protocol) ประกอบไปด้วยโปรแกรมที่ทำงานบนพื้นฐานของ transmission control protocol (TCP) โดยใช้ช่องทางการสื่อสาร (port) ที่เฉพาะเจาะจง คือ ทีซีพี พอร์ท 79 (TCP port 79) ซึ่งเรียกเป็นมาตรฐานว่า ฟิงเกอร์พอร์ท (finger port) (Zimmerman,1991) การทำงานจะประกอบด้วย 2 โปรแกรม ทำงานร่วมกันในลักษณะของผู้ขอรับบริการ และผู้ให้บริการ (client-server) โดยโปรแกรมในส่วนของผู้ขอรับบริการคือ โปรแกรมฟิงเกอร์ (finger program) หรือเรียกว่า ฟิงเกอร์ไคลเอ็นต์โปรแกรม (finger client program) และโปรแกรมในส่วนของผู้ให้บริการคือ โปรแกรมฟิงเกอร์เซอร์ฟเวอร์ (finger server program) หรือ ฟิงเกอร์เดมอน (finger daemon program, fingerd) ทำหน้าที่ร่วมกันในการให้ข้อมูลเกี่ยวกับผู้ใช้

ในการใช้งานโปรแกรมฟิงเกอร์เพื่อขอข้อมูลผู้ใช้นั้น ผู้ที่ต้องการทราบข้อมูลเพียงแต่เรียกใช้โปรแกรมฟิงเกอร์ไคลเอ็นต์เท่านั้น ถ้าหากว่าเป็นการขอข้อมูลจากเครื่องเดียวกัน โปรแกรมฟิงเกอร์จะทำการหาข้อมูล จากฐานข้อมูลต่าง ๆ ของระบบด้วยตนเอง แต่ถ้าหากเป็นการขอข้อมูลจากเครื่องอื่น ผ่านระบบเครือข่าย โปรแกรมฟิงเกอร์ไคลเอ็นต์ จะส่งคำขอข้อมูลผ่านไปยังโปรแกรมฟิงเกอร์เซอร์ฟเวอร์ ที่รอรับคำขออยู่บนเครื่องดังกล่าว แล้วโปรแกรมฟิงเกอร์ไคลเอ็นต์ จะรอรับข้อมูลที่ได้กลับมาจากฟิงเกอร์เซอร์ฟเวอร์ นำมาแสดงผลให้ผู้ที่ขอข้อมูลต่อไป

การทำงานของฟิงเกอร์ไคลเอ็นต์โปรแกรมสามารถแยกออกได้เป็น 2 ส่วน คือ

1. เมื่อมีการสอบถามข้อมูลผู้ใช้ที่อยู่บนเครื่องเดียวกัน

กรณีนี้ผู้ใช้ที่ขอข้อมูล ทำการขอข้อมูลโดยใช้ โปรแกรมฟิงเกอร์ไคลเอ็นต์ ที่มีอยู่บนเครื่องของตนเอง เรียกขอข้อมูลจากระบบของตนเอง โดยใช้รูปแบบของคำสั่งดังนี้

finger [-parameter] [username]

โดยปรกติแล้ว ถ้าในคำสั่งไม่มีการระบุชื่อผู้ใช้ (username) แล้ว โปรแกรมฟิงเกอร์จะให้ข้อมูลเกี่ยวกับ ผู้ใช้ทุก ๆ คนที่กำลังเข้าใช้งานระบบอยู่ในขณะนั้น ๆ คือ

- ชื่อเข้าใช้ระบบ (login name)
- ชื่อจริง (full name)
- หมายเลขของเทอร์มินอลที่ใช้งานอยู่ (terminal name)
- เวลาที่เริ่มเข้าใช้ระบบ (login time)
- เวลาที่ละการทำงานจากเทอร์มินอลไป (idle time)
- ที่อยู่ของผู้ใช้ (office)

ข้อมูลที่ได้จากโปรแกรมฟิงเกอร์ของเครื่องที่ใช้ระบบปฏิบัติการต่าง ๆ กัน อาจมีความแตกต่างกันได้บ้างเล็กน้อย เนื่องจากความแตกต่างของโปรแกรมฟิงเกอร์ และฐานข้อมูลผู้ใช้ระบบที่ไม่เหมือนกันในแต่ละเครื่อง ตัวอย่างของการใช้คำสั่ง finger จะได้ข้อมูลผลลัพธ์ดังต่อไปนี้

Login	Name	TTY	Idle	When	Office
ning	Chaya Limchitti	pts/10	6	Tue 08:43	
g35css	Chavalit Srisathapor	*nty/02e		Tue 08:52	
oisjc	Saint John's College	*nty/02a		Tue 08:49	
huang	Supaporn Chaithammap	pts/11	5	Tue 08:56	
u34sdk	Sataya Dankasemsant	*nty/03d		Tue 08:57	
fphacrt	Cheocharn Ratanamaha	*nty/03f		Tue 08:57	

แต่ถ้าหากว่ามีการระบุชื่อผู้ใช้ (username) แล้วโปรแกรมฟิงเกอร์ จะค้นหาข้อมูลเฉพาะที่เป็นข้อมูลผู้ใช้คนที่ระบุ โดยมีข้อมูลเช่นเดียวกับเมื่อไม่ระบุชื่อผู้ใช้ แต่จะมีข้อมูลที่เพิ่มขึ้นมาคือ

- ไดเรกทอรี่บ้าน (home directory) ของผู้ใช้
- เชลล์เริ่มต้น (startup shell) ของผู้ใช้
- ข้อมูลบรรทัดแรกในแฟ้มข้อมูล .project ที่อยู่ใน ไดเรกทอรี่บ้าน ของผู้ใช้
- ข้อมูลในแฟ้มข้อมูล .plan ที่อยู่ใน ไดเรกทอรี่บ้าน ของผู้ใช้

โดยมีลักษณะข้อมูลที่ได้ดังต่อไปนี้

Login name: g35css (messages off) In real life: Chavalit Srisathapornphat
Directory: /usr2/users/g35css Shell: /usr/bin/ksh
On since Aug 11 19:33:01 on nty/02e
Project: Super-Finger on chulkn
Plan:
Study in TCP/IP... and color printing on metal.

Engineering Computer Center.
Chulalongkorn University.
E-mail : g35css@chulkn.chula.ac.th,
lit@engecc11.eng.chula.ac.th
---

2. เมื่อมีการสอบถามข้อมูลผู้ใช้ที่อยู่ต่างเครื่องกัน

กรณีนี้ผู้ใช้เรียกใช้ โปรแกรมฟิงเกอร์ไคลเอ็นต์ บนเครื่องของตนเอง ให้ส่งคำขอข้อมูลไปยังเครื่องอื่น ๆ ที่ต้องการ โดยระบุ ชื่อของเครื่องที่ตนเองต้องการข้อมูลต่อท้ายคำขอข้อมูลตามแบบแรก คือมีรูปแบบของคำสั่งดังนี้

finger [-parameter] [username][@hostname]

ในการทำงาน โปรแกรมฟิงเกอร์จะตรวจสอบทราบว่าเป็นคำขอข้อมูล ของระบบตนเอง หรือของระบบอื่นได้ด้วยการตรวจดูว่ามีการระบุชื่อเครื่องต่อท้ายคำขอมาหรือไม่ เมื่อพบว่าเป็นคำขอข้อมูลไปยังเครื่องอื่น จะทำการส่งต่อคำขอดังกล่าวไปยังโปรแกรมฟิงเกอร์เซอร์เวอร์บนเครื่องที่ได้ระบุไว้ โดยการเปิดช่องทางการสื่อสาร ไปยังโปรแกรมฟิงเกอร์เซอร์เวอร์ ที่ช่องทางการสื่อสารแบบ ทีซีพี หมายเลข 79 แล้วรอรับผลลัพธ์ที่ได้ นำกลับมาแสดงผลให้ผู้ใช้ทราบต่อไป โดยรูปแบบของผลลัพธ์ที่ได้ มีความแตกต่างกันไป ขึ้นอยู่กับการทำงานของโปรแกรมฟิงเกอร์ของแต่ละเครื่อง

โปรแกรมฟิงเกอร์เซอร์ฟเวอร์

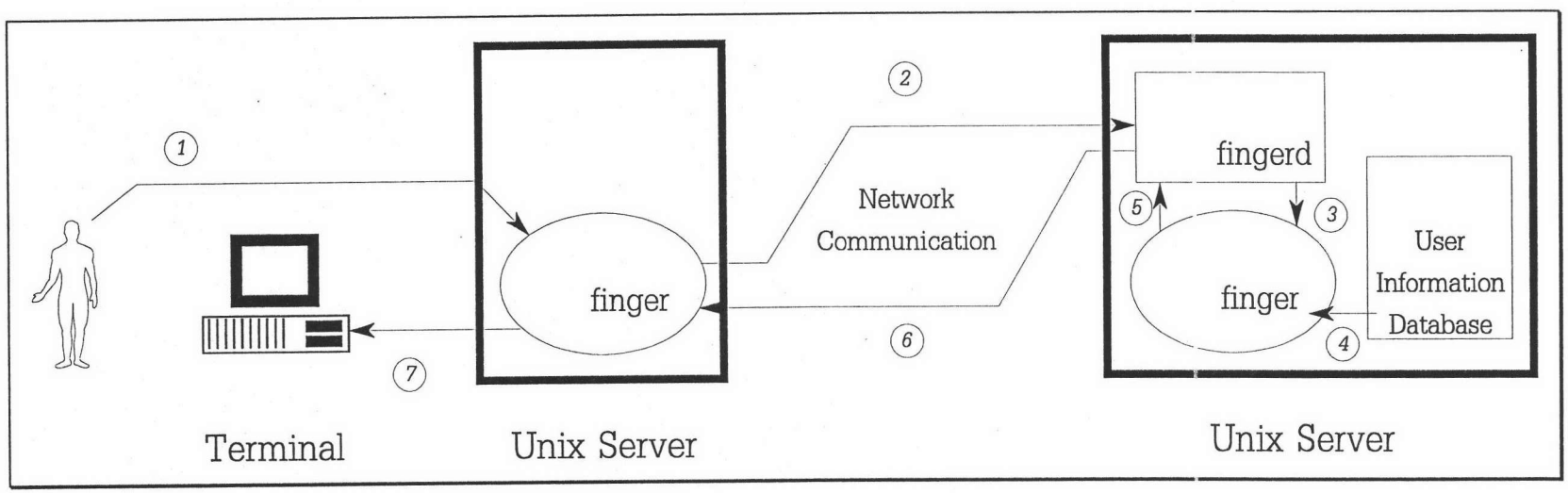
โปรแกรมฟิงเกอร์เซอร์ฟเวอร์ ทำงานในลักษณะเป็นเซอร์ฟเวอร์ โดยทำการเปิดช่องทางการสื่อสารที่ tcp port no.79 แล้วรอรับการติดต่อจากไคลเอ็นต์ จนกระทั่งมีการติดต่อมาจากไคลเอ็นต์ จึงทำการอ่านคำขอข้อมูลจากช่องทางการสื่อสารดังกล่าวเข้ามา 1 บรรทัด แล้วส่งต่อคำขอที่ได้รับมา ให้กับโปรแกรมฟิงเกอร์ ในระบบของตนเองทำการค้นหาข้อมูลให้ เมื่อได้ข้อมูลผลลัพธ์แล้วจึงส่งกลับไปให้กับไคลเอ็นต์ แล้วทำการปิดช่องทางการสื่อสาร

การทำงานของระบบฟิงเกอร์ ได้แสดงไว้ในรูปที่ 2.1

จากรูปที่ 2.1 อธิบายขั้นตอนการทำงานของระบบฟิงเกอร์เดิมได้ดังนี้

1. ผู้ใช้ส่งคำขอให้กับโปรแกรมฟิงเกอร์ไคลเอ็นต์ (finger) ผ่านทางเทอร์มินัล ของเครื่องระบบยูนิกซ์
2. โปรแกรมฟิงเกอร์ไคลเอ็นต์ ส่งต่อคำขอของผู้ใช้ไปยังโปรแกรมฟิงเกอร์เซอร์ฟเวอร์ (fingerd) บนเครื่องที่ระบุ
3. โปรแกรมฟิงเกอร์เซอร์ฟเวอร์ รับคำขอและเรียกโปรแกรมฟิงเกอร์บนเครื่องของตนเองให้ทำการค้นหาข้อมูลให้
4. โปรแกรมฟิงเกอร์ไคลเอ็นต์ทำการดึงข้อมูลจากฐานข้อมูลผู้ใช้ระบบ
5. และส่งต่อผลลัพธ์ที่ได้ให้กับโปรแกรมฟิงเกอร์เซอร์ฟเวอร์
6. โปรแกรมฟิงเกอร์เซอร์ฟเวอร์ส่งผลลัพธ์ที่ได้ผ่านระบบเครือข่ายกลับไปยังโปรแกรมฟิงเกอร์ไคลเอ็นต์บนเครื่องของผู้ใช้
7. ผู้ใช้ได้รับผลลัพธ์ทางเทอร์มินอล

ในบทต่อไปเป็นการกล่าวถึง ข้อจำกัดของโปรแกรมฟิงเกอร์ และหลักการในการขยายสมรรถนะของโปรแกรมฟิงเกอร์



รูปที่ 2.1 แสดงลักษณะการทำงานของระบบฟิงเกอร์เดิม